

AWS - Install and Configure CLI

Lesson

Downloads

Cloud Resources

The AWS Command Line Interface (AWS CLI) is a command-line tool that allows you to interact with AWS services using commands in your terminal/command prompt.

AWS CLI enables you to run commands to provision, configure, list, delete resources in the AWS cloud. Before you run any of the [aws commands](#), you need to follow three steps:

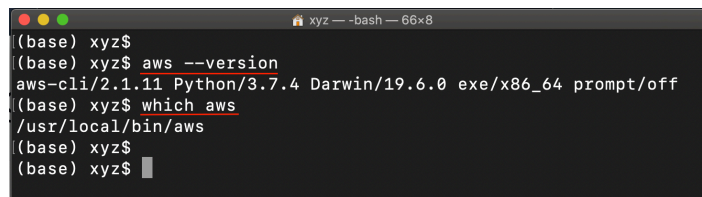
1. Install AWS CLI
2. Create an IAM user with Administrator permissions
3. Configure the AWS CLI

Step 1. Install AWS CLI v2

Refer to the official [AWS instructions to install/update AWS CLI](#) (version 2) based on your underlying OS. You can verify the installation using the following command in your terminal (macOS)/cmd (Windows).

```
# Display the folder that contains the symlink to the aws cli tool
which aws
# See the current version
aws --version
```

See the sample output below. Note that the exact version of AWS CLI and Python may vary in your system.



```
(base) xyz$
(base) xyz$ aws --version
aws-cli/2.1.11 Python/3.7.4 Darwin/19.6.0 exe/x86_64 prompt/off
(base) xyz$ which aws
/usr/local/bin/aws
(base) xyz$
(base) xyz$
```

Mac/Linux/Windows: Verify the successful installation of AWS CLI 2

[Show Transcript](#) [Summarize Video](#)

Important

- **Note 1:** The default region is not always **us-east-2** as shown on the video. You need to set the default region to the region of your AWS account. To do this, you need to take note of the region next to your user account on the AWS console:
- **Note 2:** The **AdministratorAccess** permissions policy you attach with the IAM user will automatically be removed after the session ends in ~4 hours. In that case, you will encounter permission-related error while running the aws commands in your terminal. To fix the issue, refresh the classroom page, and attach the IAM permissions policy again to the IAM user.



Region name of your AWS account

And then find the appropriate **Code** from the region mapping on this page: [AWS: Regions and Zones](#).

Step 2. Create an IAM user

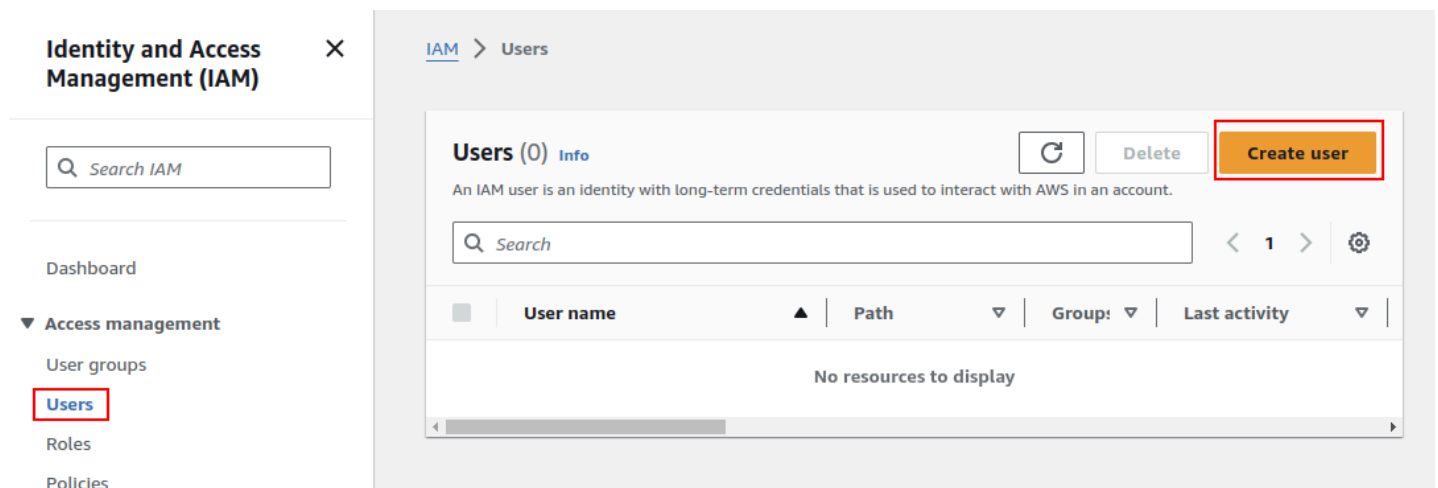
In this step, you will create an IAM user with Administrator permissions who is allowed to perform *any* action in your AWS account, only through CLI. After creating such an IAM user, we will use its **Access key** (long-term credentials)** *to* configure the AWS CLI locally.

Let's create an [AWS IAM](#) user, and copy its Access key.

AWS Identity and Access Management (IAM) service allows you to authorize users / applications (such as AWS CLI) to access AWS resources.

The Access key is a combination of an **Access Key ID** and a **Secret Access Key**. Let's see the steps to create an IAM user, and generate its Access key.

- Navigate to the [IAM Dashboard](#), and create an IAM user.



Add a new IAM user

- Set the user name, and click **Next**. **DO NOT** check **Provide user access to the AWS Management Console - optional**.

Specify user details

User details

User name

test-conupdate-3139

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Do not check this option as you don't need Management Console Access

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel

Next

Set User name.

- Set the permissions to the new user by attaching the AWS Managed **AdministratorAccess** policy from the list of existing policies.

Permissions options

☐ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1175)

Choose one or more policies to attach to your new user.

Filter by Type

×

All types

38 matches

☒

+

AdministratorAccess

AWS managed - job function

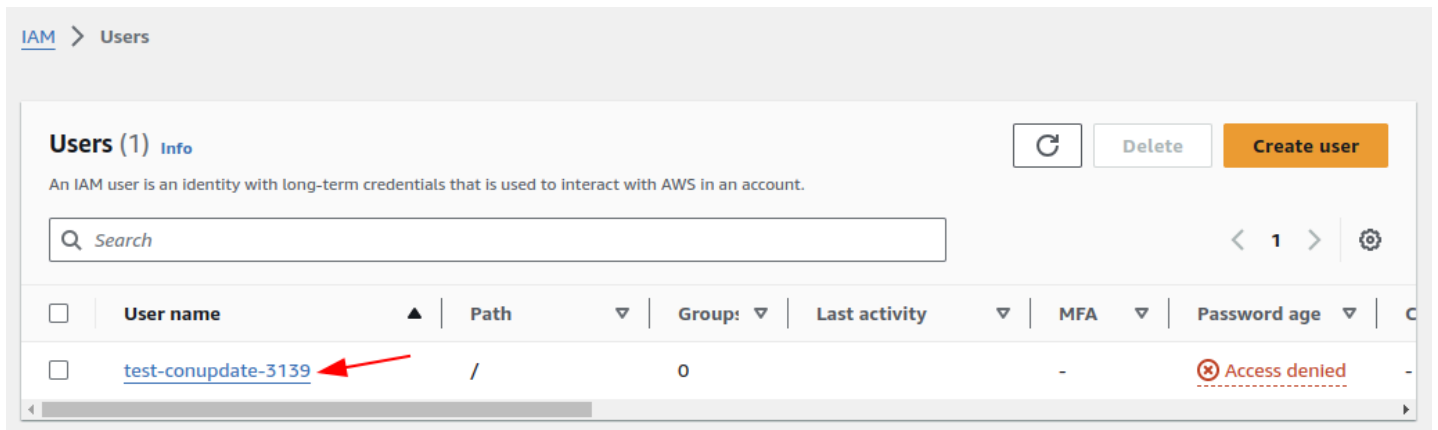
0

Attach the *AdministratorAccess* policy from the list of pre-created policies

- Provide tags [optional] ([what are tags in AWS?](#)), review the details of the new user, and finally create the new user.
- After a user is created successfully, click on the User name.

<https://learn.udacity.com/nanodegrees/nd087/parts/cd1898/lessons/de391389-41ff-416a-a242-91a8b4e80e4f/concepts/pg024...>

3/11



The screenshot shows the AWS IAM console 'Users' page. At the top, there's a breadcrumb 'IAM > Users'. Below it, the 'Users (1) Info' section includes a description: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' To the right of this section are buttons for 'Refresh', 'Delete', and 'Create user'. Below the description is a search bar and a pagination control showing '1' of 1 items. A table lists the user details:

<input type="checkbox"/>	User name	Path	Group	Last activity	MFA	Password age	Actions
<input type="checkbox"/>	test-conupdate-3139	/	0		-		⊗ Access denied

Select the created user.

- Ignore **AWS Management** Console-related warnings. Since you only need programmatic access, this can be ignored. Go to **Security Credentials** and select **Create access key**.

Summary

ARN
arn:aws:iam::468416635228:user/test-conupdate-3139

Created
January 21, 2024, 17:04 (UTC+07:00)

Console access
Access denied

Last console sign-in
Access denied

Access key 1
[Create access key](#)

Ignore AWS Management Console related warnings

Select "Security Credentials"

PermissionsGroupsTagsSecurity credentialsAccess Advisor

Console sign-in

Enable console access

Console sign-in link
https://468416635228.signin.aws.amazon.com/console

Console password
Access denied
You don't have permission to `iam:GetLoginProfile`. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)

User: arn:aws:sts::468416635228:assumed-role/voclabs/user1538874=3453428542
Service: iam
Action: GetLoginProfile
On resource(s): test-conupdate-3139

Copy

Multi-factor authentication (MFA) (0)

RemoveResyncAssign MFA device

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. [Learn more](#)

Device type	Identifier	Certifications	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment			
<div>Assign MFA device</div>			

Access keys (0)

Create access key

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

Create access key

Create Access Key for the user.

- Select **Command Line Interface (CLI)** and click **Next**.



Step 1

Access key best practices & alternatives

Step 2 - optional

Set description tag

Step 3

Retrieve access keys

Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

2. Select this checkbox

1. Select CLI

3. Click Next

Cancel

Next

Select *Command Line Interface (CLI)*

- Optional - Set description tag and click **Create access key**.

[IAM](#) > [Users](#) > [test-conupdate-3139](#) > [Create access key](#)

Step 1

[Access key best practices & alternatives](#)

Step 2 - optional

Set description tag

Step 3

Retrieve access keys

Set description tag - optional

The description for this access key will be attached to this user as a tag and shown alongside the access key.

[Optional] Set tag

Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

Click here

Cancel

Previous

Create access key

Optional - Set description tag for the access keys

- Copy the created **Access key**, **Secret access key** and store it for later use. You can also download these as a .csv file.

Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Users > test-conupdate-3139 > Create access key

Step 1

Access key best practices & alternatives

Step 2 - optional

Set description tag

Step 3


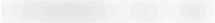


Retrieve access keys

Retrieve access keys

Copy "Access Key" and "Secret access key"

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
 	 

Hide

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [Best practices for managing AWS access keys](#).

Select this to download .csv file containing your "access key" and "secret access key"

Click Done.

Download .csv file

Done

Copy Access Key and Secret Access Key

Note that you can generate a temporary Access key in the classroom as well, as shown in the snapshot below. But, the classroom generated access key is valid for a for a few hours only.

Notice that it has a session token associated with it. The session token needs to be included in your CLI configuration for it to be usable. This will be discussed in more detail in the next section below.

AWS Access Key ID

ASIAXRXYGSKPOIMFHY2

AWS Secret Access Key

+GDNeKipgOv2KK+oGVCsKhHUEds1rZZWtDzhQMgs

AWS Session Token

FwoGZXlvYxdzEF8aDNkzzGGBEjKp6o6tCLVAaKtGMBZGrHOX+YWbJOQOsn3rVjt7h2IT6zB
oJrC4+DfPiiA3m6SzwUdN4L939akZlbQOy114wjsdw2OpYaFP3L9bALE0ln2HlffgUqlcSzk5F
TFCu7idaY6zwFselqjRRbqWzjAdqfgxvQu0PbBj/y6l8Fu0Hm1xDzGwFRRMx02hkD2K87Ckps
qQenm+G2pV4dpXELQzGG+TtLHEZajHjY7RMW30O96P/8oC7NzlrAtF9/H3ZNOGuLkvwzy6
uWeZP45fC0/M6jx30xcuQKOScX6RUXkiicie2TBjltB0PUzIF66VqsjANaE0aZavuVMKAjuoR3X
5ZAMsrY65EgIKlulV7FqAfeNbU/

OPEN CLOUD CONSOLE

Access key shown in the classroom after clicking on the "OPEN CLOUD GATEWAY" button

Step 3. Configure the AWS CLI

You will need to configure the following four items on your local machine before you can interact with any of the AWS services:

1. **Access key** - It is a combination of an *Access Key ID* and a *Secret Access Key*. Together, they are referred to as *Access key*. You can generate an Access key from the AWS IAM service, and specify the level of permissions (authorization) with the help of *IAM Roles*.
2. **Default AWS Region** - It specifies the AWS Region where you want to send your requests by default.
3. **Default output format** - It specifies how the results are formatted. It can either be a json, yaml, text, or a table.
4. **Profile** - A collection of settings is called a profile. The default profile name is `default`, however, you can create a new profile using the `aws configure --profile new_name` command.

Here are the steps to configure the AWS CLI in your terminal:

- Run the command below to configure the AWS CLI using the *Access Key ID* and a *Secret Access Key* generated in the previous step. If you have closed the web console that showed the access key, you can open the downloaded access key file (.csv) to copy the keys later.

```
aws configure
```

If you already have a profile set locally, you can use `--profile <profile-name>` option with any of the AWS commands above. This will resolve the conflict with the existing profiles set up locally. Next, use the following values in the prompt that would appear:

Important

If you are using the Udacity-generated Access key, you should set the `aws_session_token` parameter.

```
aws configure set aws_session_token "XXXXXXX"
```

where `"XXXXXXX"` is the **AWS Session Token** copied from the classroom after clicking on the "OPEN CLOUD GATEWAY" button.

If you are using the Access key of an Admin IAM user, you do not need to set this parameter.

- The commands above will store the access key in a default file `~/.aws/credentials` and store the profile in the `~/.aws/config` file. Upon prompt, paste the copied access key (access key id and secret access key). Enter the default region as `us-east-1` and output format as `json`. You can verify the saved config using:


```

udacity@Udacitys-MacBook-Pro:~
(base) → cat ~/.aws/credentials
[default]
aws_access_key_id = [REDACTED]/BX7U
aws_secret_access_key = [REDACTED]DHHd
(base) → cat ~/.aws/config
[default]
region = us-east-1
output = json
(base) → aws configure list
      Name                               Value                                Type      Location
      ----                               -
      profile                            <not set>                           None      None
      access_key                         *****BX7U                         shared-credentials-file
      secret_key                         *****DHHd                         shared-credentials-file
      region                             us-east-1                           config-file  ~/.aws/config
(base) → aws configure list-profiles
default
(base) → aws configure set region us-east-1
(base) → aws configure set aws_session_token "" --profile default
(base) → aws iam list-users

```

Mac/Linux: A successful configuration

The above steps should be sufficient for setting up your AWS CLI locally, but if you still can't run AWS commands, you may try following the steps below.

- **Mac/Linux users** Let the system know that your sensitive information is residing in the .aws folder

```

export AWS_CONFIG_FILE=~/.aws/config
export AWS_SHARED_CREDENTIALS_FILE=~/.aws/credentials

```

- **Windows users with GitBash only** You will have to set the environment variables. Run the following commands in your GitBash terminal:

```

setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
setx AWS_DEFAULT_REGION us-east-1

```

Replace the access key ID and secret, as applicable to you. Windows users using WSL do not need this step, they will follow all steps as if they are Linux users.

```

MINGW64/c/Users/sudha
sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ aws configure --profile default
AWS Access Key ID [*****p2MM]: *****
AWS Secret Access Key [*****sZ+F]: *****
Default region name [us-east-1]: us-east-1
Default output format [json]:

sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ setx AWS_ACCESS_KEY_ID *****
SUCCESS: Specified value was saved.

sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ setx AWS_SECRET_ACCESS_KEY *****
SUCCESS: Specified value was saved.

sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ setx AWS_DEFAULT_REGION us-east-1
SUCCESS: Specified value was saved.

sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "UserName": "Admin",
      "UserId": "AIDAZMHSC3LYUB5ODMUNS",
      "Arn": "arn:aws:iam:644752792305:user/Admin",
      "CreateDate": "2021-03-09T11:31:13+00:00"
    },
    {
      "Path": "/",
      "UserName": "windowsAdmin",
      "UserId": "AIDAZMHSC3LYRQFEF5H5Z",
      "Arn": "arn:aws:iam:644752792305:user/windowsAdmin",
      "CreateDate": "2021-03-09T11:54:01+00:00"
    }
  ]
}

```

Windows: Successful configuration using the GitBash terminal

Step 4. Run your first AWS CLI command

- Check the successful configuration of the AWS CLI, by running either of the following AWS command:

```

# If you've just one profile set locally
aws iam list-users

# If you've multiple profiles set locally
aws iam list-users --profile <profile-name>

```

The output will display the details of the recently created user:

```

{
  "Users": [
    {
      "Path": "/",
      "UserName": "Admin",
      "UserId": "AIDAZMXYZ3LY2BNC5ZM5E",
      "Arn": "arn:aws:iam::388752792305:user/Admin",
      "CreateDate": "2021-01-28T13:44:15+00:00"
    }
  ]
}

```

Troubleshoot

If you are facing issues while following the commands above, refer to the detailed instructions here -

- [1. Configuration basics](#)
- [2. Configuration and credential file settings](#)

3. [Environment variables to configure the AWS CLI](#)

Updating the specific variable in the configuration

In the future, you can set a single value, by using the command, such as:

```
# Syntax
# aws configure set <varname> <value> [--profile profile-name]
aws configure set default.region us-east-1
```

It will update only the region variable in the existing default profile.